PORTAL
USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

cache miss detect SEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used **cache miss detect** Found **18,999** of **161,645**

Sort results by: relevance

Display results: expanded form

Save results to a Binder
Search Tips
☐ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next
Best 200 shown

Relevance scale □ ▭ ▬ ■ ■

1 Cache: Reducing traffic generated by conflict misses in caches ■
Pepijn J. de Langen, Ben Juurlink
April 2004 **Proceedings of the 1st conference on Computing frontiers**

Full text available: pdf(246.59 KB) Additional Information: full citation, abstract, references, citings, index terms

Off-chip memory accesses are a major source of power consumption in embedded processors. In order to reduce the amount of traffic between the processor and the off-chip memory as well as to hide the memory latency, nearly all embedded processors have a cache on the same die as the processor core. Because small caches dissipate less power and are cheaper than large caches, a small cache is preferable to a large cache. Furthermore, because set-associative caches consume more power than direct-mapp ...

Keywords: caches, conflict misses, embedded processors, power reduction

2 Memory-wall: Bloom filtering cache misses for accurate data speculation and prefetching ■
Jih-Kwon Peir, Shih-Chang Lai, Shih-Lien Lu, Jared Stark, Konrad Lai
June 2002 **Proceedings of the 16th international conference on Supercomputing**

Full text available: pdf(248.57 KB) Additional Information: full citation, abstract, references, citings, index terms

A processor must know a load instruction's latency to schedule the load's dependent instructions at the correct time. Unfortunately, modern processors do not know this latency until well after the dependent instructions should have been scheduled to avoid pipeline bubbles between themselves and the load. One solution to this problem is to predict the load's latency, by predicting whether the load will hit or miss in the data cache. Existing cache hit/miss predictors, however, can only correctly ...

Keywords: bloom filter, data cache, data prefetching, data speculation, instruction scheduling

3 Avoiding conflict misses dynamically in large direct-mapped caches ■
Brian N. Bershad, Dennis Lee, Theodore H. Romer, J. Bradley Chen
November 1994 **Proceedings of the sixth international conference on Architectural support for programming languages and operating systems**, Volume 29 ,

28 Issue 11 , 5

Full text available: pdf(1.37 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes a method for improving the performance of a large direct-mapped cache by reducing the number of conflict misses. Our solution consists of two components: an inexpensive hardware device called a Cache Miss Lookaside (CML) buffer that detects conflicts by recording and summarizing a history of cache misses, and a software policy within the operating system's virtual memory system that removes conflicts by dynamically remapping pages whenever large numbers of conflict miss ...

4 The detection and elimination of useless misses in multiprocessors

Michel Dubois, Jonas Skeppstedt, Livio Ricciulli, Krishnan Ramamurthy, Per Stenström

May 1993 **ACM SIGARCH Computer Architecture News , Proceedings of the 20th annual international symposium on Computer architecture**, Volume 21 Issue 2

Full text available: pdf(1.03 MB) Additional Information: full citation, abstract, references, citings, index terms

In this paper we introduce a new classification of misses in shared-memory multiprocessors based on interprocessor communication. We identify the set of essential misses, i.e., the smallest set of misses necessary for correct execution. Essential misses include cold misses and true sharing misses. All other misses are useless misses and can be ignored without affecting the correctness of program execution. Based on the new classification we compare the effectiveness of five different protoc ...

5 Reducing cache misses using hardware and software page placement

Timothy Sherwood, Brad Calder, Joel Emer

May 1999 **Proceedings of the 13th international conference on Supercomputing**

Full text available: pdf(1.50 MB) Additional Information: full citation, references, citings, index terms

6 DSTRIDE: data-cache miss-address-based stride prefetching scheme for multimedia processors

Hariprakash. G, Achutharaman. R, Amos R. Omondi

January 2001 **Australian Computer Science Communications , Proceedings of the 6th Australasian conference on Computer systems architecture ACSAC '01**, Volume 23 Issue 4

Full text available: pdf(928.14 KB)
Publisher Site

Additional Information: full citation, abstract, references

Prefetching reduces cache miss latency by moving data up in memory hierarchy before they are actually needed. Recent hardware-based stride prefetching techniques mostly rely on the processor pipeline information (e.g. program counter and branch prediction table) for prediction. Continuing developments in processor microarchitecture drastically change core pipeline design and require that existing hardware-based stride prefetching techniques be adapted to the evolving new processor architectures. ...

7 Simple compiler algorithms to reduce ownership overhead in cache coherence protocols

Jonas Skeppstedt, Per Stenström

November 1994 **Proceedings of the sixth international conference on Architectural support for programming languages and operating systems**, Volume 29 , 28 Issue 11 , 5

Full text available: pdf(1.47 MB) Additional Information: full citation, abstract, references, citings, index terms

We study in this paper the design and efficiency of compiler algorithms that remove ownership overhead in shared-memory multiprocessors with write-invalidate protocols. These algorithms detect loads followed by stores to the same address. Such loads are marked and constitute a hint to the cache to obtain an exclusive copy of the block. We consider three algorithms where the first one focuses on load-store sequences within each basic block of code and the other two analyse the existence of l ...

8 Predicting data cache misses in non-numeric applications through correlation profiling
Todd C. Mowry, Chi-Keung Luk
December 1997 **Proceedings of the 30th annual ACM/IEEE international symposium on Microarchitecture**
Full text available: pdf(876.36 KB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

To maximize the benefit and minimize the overhead of software-based latency tolerance techniques, we would like to apply them precisely to the set of dynamic references that suffer cache misses. Unfortunately, the information provided by the state-of-the-art cache miss profiling technique (summary profiling) is inadequate for references with intermediate miss ratios - it results in either failing to hide latency, or else inserting unnecessary overhead. To overcome this problem, we propose and ev ...

Keywords: profiling, cache miss prediction, correlation, non-numeric applications, latency tolerance.

9 Owner prediction for accelerating cache-to-cache transfer misses in a cc-NUMA architecture
Manuel E. Acacio, José González, José M. García, José Duato
November 2002 **Proceedings of the 2002 ACM/IEEE conference on Supercomputing**
Full text available: pdf(120.57 KB) Additional Information: full citation, abstract, references, citings, index terms

Cache misses for which data must be obtained from a remote cache (cache-to-cache transfer misses) account for an important fraction of the total miss rate. Unfortunately, cc-NUMA designs put the access to the directory information into the critical path of 3-hop misses, which significantly penalizes them compared to SMP designs. This work studies the use of owner prediction as a means of providing cc-NUMA multiprocessors with a more efficient support for cache-to-cache transfer misses. Our propo ...

10 Memory systems: Cluster miss prediction with prefetch on miss for embedded CPU instruction caches
Ken Batcher, Robert Walker
September 2004 **Proceedings of the 2004 international conference on Compilers, architecture, and synthesis for embedded systems**
Full text available: pdf(343.66 KB) Additional Information: full citation, abstract, references, index terms

Soft CPU cores are often used in embedded systems, yet they limit opportunities to improve cache performance to hardware assistance outside the CPU core. Instruction prefetching is commonly used, but the popular Prefetch On Miss (POM) technique is less helpful when the instruction flow does not follow a sequential execution order, which is often the case in real-time networking applications. Cluster Miss Prediction (CMP) can help in those worst case situations when cache misses do not follow a s ...

Keywords: WCET, cache design, cache prefetch, embedded systems, hiding memory latency, networking

11 Run-time spatial locality detection and optimization
Teresa L. Johnson, Matthew C. Merten, Wen-Mei W. Hwu
December 1997 **Proceedings of the 30th annual ACM/IEEE international symposium on Microarchitecture**
Full text available: pdf(1.11 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

As the disparity between processor and main memory performance grows, the number of execution cycles spent waiting for memory accesses to complete also increases. As a result, latency hiding techniques are critical for improved application performance on future processors. We present a microarchitecture scheme which detects and adapts to varying spatial locality, dynamically adjusting the amount of data fetched on a cache miss. The Spatial Locality Detection Table, introduced in this paper, faci ...

Keywords: data cache, cache management, spatial locality, prefetching, block size

12 The Performance of Runtime Data Cache Prefetching in a Dynamic Optimization System
Jiwei Lu, Howard Chen, Rao Fu, Wei-Chung Hsu, Bobbie Othmer, Pen-Chung Yew, Dong-Yuan Chen
December 2003 **Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture**
Full text available: pdf(253.79 KB) Additional Information: full citation, abstract, citings, index terms

Traditional software controlled data cache prefetching isoften ineffective due to the lack of runtime cache miss andmiss address information. To overcome this limitation, weimplement runtime data cache prefetching in the dynamicoptimization system ADORE (ADaptive Object code RE-optimization).Its performance has been compared withstatic software prefetching on the SPEC2000 benchmarksuite. Runtime cache prefetching shows better performance.On an Itanium 2 based Linux workstation, it can increasepe ...

13 Improving data cache performance by pre-executing instructions under a cache miss
James Dundas, Trevor Mudge
July 1997 **Proceedings of the 11th international conference on Supercomputing**
Full text available: pdf(1.04 MB) Additional Information: full citation, references, citings, index terms

14 Optimizing cache miss equations polyhedra
Nerina Bermudo, Xavier Vera, Antonio González, Josep Llosa
March 2000 **ACM SIGARCH Computer Architecture News**, Volume 28 Issue 1
Full text available: pdf(744.61 KB) Additional Information: full citation, abstract, index terms

Cache Miss Equations (CME) [GMM97] is a method that accurately describes the cache behavior by means of polyhedra. Even though the computation cost of generating CME is a linear function of the number of references, to solve them is a very time consuming task and thus trying to study a whole program may be infeasible.In this work, we present effective techniques that exploit some properties of the particular polyhedra generated by CME. Such technique reduce the complexity of the algorithm to sol ...

15 A large, fast instruction window for tolerating cache misses
Alvin R. Lebeck, Jinson Koppanalil, Tong Li, Jaidev Patwardhan, Eric Rotenberg
May 2002 **ACM SIGARCH Computer Architecture News**, Volume 30 Issue 2

Full text available: pdf(1.22 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

Instruction window size is an important design parameter for many modern processors. Large instruction windows offer the potential advantage of exposing large amounts of instruction level parallelism. Unfortunately naively scaling conventional window designs can significantly degrade clock cycle time, undermining the benefits of increased parallelism.This paper presents a new instruction window design targeted at achieving the latency tolerance of large windows with the clock cycle time of small ...

Keywords: Instruction Window, Memory Latency, Cache Memory, Latency Tolerance

16 An efficient cache-based access anomaly detection scheme

Sang L. Min, Jong-Deok Choi

April 1991 **Proceedings of the fourth international conference on Architectural support for programming languages and operating systems**, Volume 19 , 25 , 26 Issue 2 , Special Issue , 4

Full text available: pdf(1.20 MB) Additional Information: full citation, references, citings, index terms

17 Large models & large displays: Cache-oblivious mesh layouts

Sung-Eui Yoon, Peter Lindstrom, Valerio Pascucci, Dinesh Manocha

July 2005 **ACM Transactions on Graphics (TOG)**, Volume 24 Issue 3

Full text available: pdf(447.02 KB) Additional Information: full citation, abstract, references

We present a novel method for computing cache-oblivious layouts of large meshes that improve the performance of interactive visualization and geometric processing algorithms. Given that the mesh is accessed in a reasonably coherent manner, we assume no particular data access patterns or cache parameters of the memory hierarchy involved in the computation. Furthermore, our formulation extends directly to computing layouts of multi-resolution and bounding volume hierarchies of large meshes.We deve ...

18 Fingerprinting: bounding soft-error detection latency and bandwidth

Jared C. Smolens, Brian T. Gold, Jangwoo Kim, Babak Falsafi, James C. Hoe, Andreas G. Nowatzyk

October 2004 **Proceedings of the 11th international conference on Architectural support for programming languages and operating systems**, Volume 39 , 32 , 38 Issue 11 , 5 , 5

Full text available: pdf(229.65 KB) Additional Information: full citation, abstract, references, index terms

Recent studies have suggested that the soft-error rate in microprocessor logic will become a reliability concern by 2010. This paper proposes an efficient error detection technique, called *fingerprinting*, that detects differences in execution across a dual modular redundant (DMR) processor pair. Fingerprinting summarizes a processor's execution history in a hash-based signature; differences between two mirrored processors are exposed by comparing their fingerprints. Fingerprinting tightly ...

Keywords: backwards error recovery (BER), dual modular redundancy (DMR), error detection, soft errors

19 Runtime identification of cache conflict misses: The adaptive miss buffer

Jamison D. Collins, Dean M. Tullsen

November 2001 **ACM Transactions on Computer Systems (TOCS)**, Volume 19 Issue 4

Full text available: pdf(1.08 MB) Additional Information: full citation, abstract, references, index terms

This paper describes the miss classification table, a simple mechanism that enables the processor or memory controller to identify each cache miss as either a conflict miss or a capacity (non-conflict) miss. The miss classification table works by storing part of the tag of the most recently evicted line of a cache set. If the next miss to that cache set has a matching tag, it is identified as a conflict miss. This technique correctly identifies 88&percnt; of misses.Several applications of this i ...

Keywords: Cache architecture, adaptive miss buffer, cache exclusion, conflict misses, prefetching, victim cache

20 Using dataflow analysis techniques to reduce ownership overhead in cache coherence protocols

Jonas Skeppstedt, Per Stenström

November 1996 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 18 Issue 6

Full text available: pdf(284.68 KB) Additional Information: full citation, abstract, references, index terms, review

In this article, we explore the potential of classical dataflow analysis techniques in removing overhead in write-invalidate cache coherence protocols for shared-memory multiprocessors. We construct the compiler algorithms with varying degree of sophistication that detect loads followed by stores to the same address. Such loads are marked and constitute a hint to the cache to obtain an exclusive copy of the block so that the subsequent store does not introduce access penalties. The simplest ...

Keywords: cache coherence, dataflow analysis, performance evaluation

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next